Chapter 2

Preliminaries

2.1 Feedforward Neural Networks

Artificial neural networks, which we will commonly refer to simply as *neural networks* or *nets*, are the fundamental mathematical objects of deep learning. They consist of an *input layer*, a number of *hidden layers*, and an *output layer*. Each layer consists of a finite number of nodes. We call the number of layers the *depth* of the network, and the number of nodes in a given layer the *width* of the layer. In the simplest case the collection of neurons form a directed acyclic graph where the subgraph generated by successive layers is fully connected, that is, there exist edges connecting each node from layer i to every node of layer $i+1$. Such a network is called a *feedforward network*. A typical schematic of a feedforward neural network is seen in Fig. 2.1. **(Fix figures)**

In general, the *architecture* of the network is the data consisting of: the number of layers L of the network, the width of each layer $(d_1, \dots, d_{L+1}) \subset \mathbb{N}_{>0}^{L+1}$ and the graph describing the connectivity of the layers of nodes. Each layer d_i connects to layer d_{i+1}, but not necessarily in a fully connected way.

Remark 1. The hidden layers refer to the number of internal layers of nodes which are not the input or output layers, thus there are $L-1$ hidden layers in a depth L network.

Remark 2. The architectures of modern neural networks often have very large depth, hence the class of such neural networks is commonly referred to as *deep learning*. Furthermore, architectures with different graphical properties have been recently used with great success, including graphs with the presence of loops (*recurrent neural networks*), or layers that are not fully connected (*convolutional neural networks*). We refer the reader to Goodfellow et al. 2016 for elaboration on such architectures. In this thesis our study will be restricted to feedforward neural networks.

To each edge between a node $i \in \{1, \dots, d_{l-1}\}$ in layer $l-1$ to a node $j \in \{1, \dots, d_l\}$ in layer l is a weight $w_{j,i}^l \in \mathbb{R}$, and to each node j there is a bias $b_j^l \in \mathbb{R}$. This gives rise to an affine function $A^l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l+1}}$, $A^l(x) = (w^l)^T x + b^l$, where the first term is matrix multiplication. Neurons are then "activated" via composition of the affine function with a vectorised activation function $\sigma(x)$, thus the output a^l from each layer l can be expressed recursively as

$$a^l = \sigma(w^l a^{l-1} + b^l)\,.$$

Interestingly, there is no widely accepted definition of an activation function, and indeed when one examines the plethora of such functions that are used in practice it is clear that there are no common traits other than the fact that they are non-zero somewhere! In

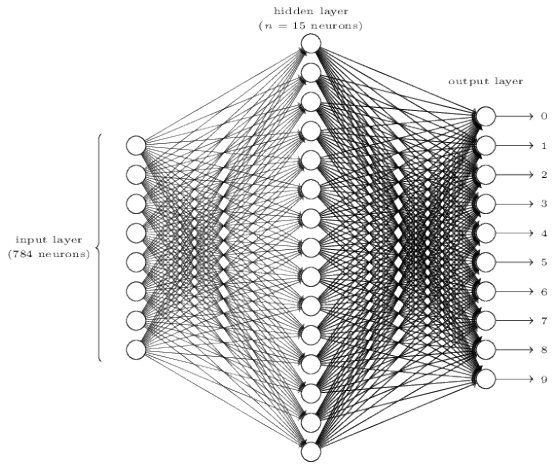

Figure 2.1: A feedforward neural network as a fully connected graph *placeholder graph*

the early literature (see (Rosenblatt 1962)) the activation function was typically the step function

$$\sigma_H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases},$$

which thus elucidates the use of the term "activation". Neural networks with $\sigma_H(x)$ as the activation function are called *perceptron* networks. Other common activation functions include the sigmoid function $\sigma_s(x) = \frac{1}{1+e^{-x}}$ and hyperbolic tangent $\sigma_t(x) = \tanh(x)$, but the one we will almost exclusively discuss throughout this thesis is the Rectified Linear Unit (ReLU) defined by

$$\text{ReLU}(x) = \max\{0, x\} = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \tag{2.1}$$

Remark 3. We may extend the definition of any of these activation functions to be vectorised by writing $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ for some n such that $\sigma(x_1, \ldots, x_n) = (\sigma(x_1), \ldots, \sigma(x_n))$. From here on we assume that any activation function mentioned has naturally been vectorised.

Remark 4. The ReLU function is not analytic at $x = 0$, which we will see is problematic when discussing such nets in the context of singular learning theory in Section 2.4. An analytic alternative to ReLU is the swish function given by

$$\sigma_\beta(x) = \frac{x}{1 + e^{-\beta x}}$$

for some $\beta \in \mathbb{R}$, which is analytic everywhere and also satisfies $\lim_{\beta \to \infty} \sigma_\beta(x) = \text{ReLU}(x)$.

We now have all of the pieces to define the neural networks we will examine in this thesis.

Definition 1. Let $W \subseteq \mathbb{R}^d$ denote the *weight space*, where d is the number of parameters. A *feedforward ReLU neural network* of depth L with widths $(d_1, \ldots, d_{L+1}) \subset \mathbb{N}_{>0}^L$ such that there are $d_1 = N$ inputs and $d_{L+1} = M$ outputs, is a feedforward neural network with activation function $\sigma(x) = \text{ReLU}(x)$. That is, it is a function

$$f : \mathbb{R}^N \times W \longrightarrow \mathbb{R}^M$$
$$f(x, w) = (A_L \circ \text{ReLU} \circ A_{L-1} \circ \text{ReLU} \circ \cdots \circ \text{ReLU} \circ A_1)(x)$$

such that for each hidden layer $l = 1, \dots, L$ there is an affine function $A_l : \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l+1}}$ paramaterised by weights $w^l \in \mathrm{Mat}_{d_l \times d_{l+1}}(\mathbb{R})$ and biases $b^l \in \mathrm{Mat}_{d_{l+1} \times 1}(\mathbb{R})$ given by

$$A_l(x) = (w^l)^T x + b^l \,.$$

In the case where w^l is a column vector we may write this as $A_l(x) = \langle w^l, x \rangle + b^l$ where $\langle \, , \rangle$ denotes the dot product.

Remark 5. We may sometimes adopt the notation $w_{\bullet,j} = (w_{1,j}, w_{2,j}, \dots, w_{d_l,j})^T$ for column vectors. Further, when w is assumed fixed we will often denote $f_w := f(\_, w) : \mathbb{R}^N \to \mathbb{R}^M$.

Feedforward ReLU networks are, by the definition of ReLU, piecewise affine functions, meaning networks of relatively low depth and widths are quite simple functions. However, this class of networks have been shown to be very expressive in the sense that, under suitable conditions, they are universal approximators of arbitrary Lebesgue integrable functions. The following Universal Approximation Theorem of Lu et al. 2017, which we shall not prove here but is included here for completeness, describes this:

Theorem. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a Lebesgue-integrable function and let $\varepsilon > 0$ be arbitrary. Then there exists a feedforward ReLU neural network $f_w : \mathbb{R}^n \to \mathbb{R}$ of some depth L with bounded widths $d_l \leq n + 4$ such that*

$$\int_{\mathbb{R}^n} |g(x) - f(x, w)| dx < \varepsilon \,.$$

This theorem thus demonstrates the power, and hence popularity, of feedforward ReLU neural networks in modern deep learning: they are simple functions to compute, yet have the ability to express very complicated functions to arbitrary precision. **Actually this is only for $M = 1$, so maybe should find a better theorem**.

We will return to the specific case of two layer networks with $M = 1$ output in Section 3.1, but for now we present a single example of such a function for ease of understanding.

Example 1. Let $f_w : \mathbb{R}^2 \to \mathbb{R}^1$ be a two layer neural network with two outputs, one input, and width $d_1 = 3$ in the hidden layer, hence has the form

$$\begin{aligned} f_w(x) = (A_2 \circ \mathrm{ReLU} \circ A_1)(x) &= \left\langle w^2, \mathrm{ReLU}((w^1)^T x + b^1) \right\rangle + b^2 \\ &= w^2_{1,1} \mathrm{ReLU}(w^1_{1,1} x_1 + w^1_{2,1} x_2 + b^1_1) + w^2_{2,1} \mathrm{ReLU}(w^1_{1,2} x_1 + w^1_{2,2} x_2 + b^1_2) \\ &\quad + w^2_{3,1} \mathrm{ReLU}(w^1_{1,3} x_1 + w^1_{2,3} x_2 + b^1_3) + b^2_1 \,. \end{aligned}$$

Consider $w \in W$ such that

$$f_w(x) = \mathrm{ReLU}\left(\frac{1}{2}x_1 + \frac{\sqrt{3}}{2}x_2 - \frac{1}{3}\right) + \mathrm{ReLU}\left(\frac{1}{2}x_1 - \frac{\sqrt{3}}{2}x_2 - \frac{1}{3}\right) + \mathrm{ReLU}\left(\frac{1}{2}x_1 + \frac{1}{3}\right) \,.$$

Once I produce the picture I should give the piecewise definition in the different regions.

2.2 The objects of statistical learning theory

Given a dataset D_n of inputs $x \in \mathbb{R}^N$ and outputs $y \in \mathbb{R}^M$, the objective of deep learning is to train a *learning machine* (or *model*) $p(y|x, w)$ to produce the ouputs for the given inputs

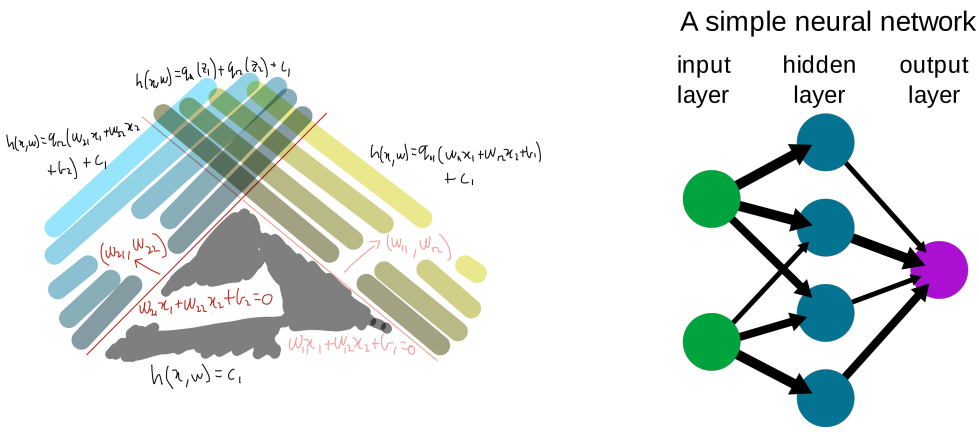

Figure 2.2: Activation boundaries and network **placeholder pic**

on the training set D_n, and at the same time generalise beyond this dataset and perform prediction tasks on data the learning machine has not seen before. This training, therefore, amounts to estimating model parameters w that minimise these quantities of interest, though they can't necessarily be achieved simultaneously due to a typical statistical trade-off (e.g. the classical bias-variance trade-off, see (Casella and Berger 2002)).

Due to computational benefits such as training parallelisation and scalability, it is standard practice within modern deep learning to view this estimation procedure within a frequentist framework, where w is viewed as being unknown yet fixed (Casella and Berger 2002). Training is then performed using the Stochastic Gradient Descent (SGD) algorithm (Goodfellow et al. 2016), which is achieved at scale via the famous backpropagation algorithm.

However, as in Watanabe's *Algebraic Geometry and Statistical Learning Theory* (Watanabe 2009), our view of the learning procedure will be within the Bayesian framework, whereby the model parameters $w \in W$ are assumed to be drawn from a probability distribution, and the learning goal thus becomes estimating the posterior distribution $p(w|D_n)$. This framework is much more theoretically tractable, elegant and illustrative.

Remark 6. An assumption within the deep learning literature is that training via SGD is approximately equivalent to sampling from a Bayesian posterior, with evidence mounting that this is indeed the case (see Mingard et al. 2020 and Mandt et al. 2018). Such a statement effectively justifies our use of Bayesian statistics in drawing conclusions about deep learning, but it is worth keeping in mind that this connection is not yet rigorous.

The following exposition of Bayesian statistics and related definitions is largely drawn from (Watanabe 2018), (Watanabe 2009) and (Casella and Berger 2002).

2.2.1 Bayesian statistics

Let $(\Omega, \mathcal{B}, \mathbb{P})$ be a probability space and $(X, Y) : \Omega \to \mathbb{R}^N \times \mathbb{R}^M$ a jointly random variable subject to the probability density $q(y, x) = q(y|x)q(x)$, where X is the input to the learning machine and Y is the output. Recall that the objective of statistical learning is to estimate the true distribution $q(y, x)$ given a collection of random samples D_n of the form

$$D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

where each (X_i, Y_i) is independent and identically distributed, thus leading to a probability density of the form

$$q((x_1, y_1), \dots, (x_n, y_n)) = q(x_1, y_1) \dots q(x_n, y_n)\,.$$

We then assume that the data is drawn according to a joint probability distribution $p(y, x|w) = p(y|x, w)q(x)$ which we call the *model*, paramaterised by some parameter $w \in W$. Since the samples are independent and identically distributed, we may define the *likelihood* function as

$$l(w|x, y) := p(y_1, \dots, y_n, x_1, \dots, x_n|w) = \prod_{i=1}^{n} p(y_i, x_i|w) = \prod_{i=1}^{n} p(y_i|x_i, w)q(x_i)\,. \tag{2.2}$$

The statistical learning goal is thus to estimate the posterior density $p(w|x, y)$ subject to a dataset D_n. Let $\varphi(w)$ denote the *prior* probability density of $w \in W$, which is a "subjective distribution based on the experimenter's belief and is formulated before the data is seen" (Casellas). By Bayes' rule, the *posterior* probability density is given by

$$p(w|D_n) := \frac{p(D_n|w)\varphi(w)}{p(D_n)} = \frac{1}{p(D_n)}\varphi(w)\prod_{i=1}^{n} p(y_i, x_i|w) = \frac{1}{p(D_n)}\varphi(w)\prod_{i=1}^{n} p(y_i|x_i, w)q(x_i)\,,$$

where the *evidence* $p(D_n)$ (also called the *marginal likelihood*) is given by

$$p(D_n) = \int_W p(D_n|w)\varphi(w)dw = \int_W \prod_{i=1}^{n} q(x_i)p(y_i|x_i, w)\varphi(w)dw\,,$$

which ensures the posterior is normalised and thus a well defined probability density. But since $\prod_{i=1}^{n} q(x_i)$, which is independent of w, is a factor of both $p(D_n|w)$ and $p(D_n)$, we may simplify this to give a more concise definition:

Definition 2. The *posterior* probability density $p(w|D_n)$ is given by

$$p(w|D_n) = \frac{1}{Z_n}\varphi(w)\prod_{i=1}^{n} p(y_i|x_i, w)\,, \quad \text{where} \quad Z_n = \int_W \varphi(w)\prod_{i=1}^{n} p(y_i|x_i, w)dw\,. \tag{2.3}$$

We call Z_n the *partition function*.

Remark 7. Clearly the partition function and evidence are related via $p(D_n) = Z_n \prod_{i=1}^{n} q(x_i)$.

Within our setup we are considering the random variable D_n (associated to the random variables the inputs X_i and the outputs Y_i), and the random variable w (which we do not denote with a capital for notational clarity). As such, for clarity we define the following expectations:

Definition 3. Let $g(X, Y)$ be a function of the random input and output variables and $f(w)$ a function of the random weight variables. We define:

$$\mathbb{E}_{D_n}[g(X, Y)] := \iint_{\mathbb{R}^{N+M}} q(x, y)g(x, y)dxdy\,,$$
$$\text{and} \qquad \mathbb{E}_w[f(w)] = \int_W p(w|D_n)f(w)dw\,.$$

For this thesis we will restrict our attention to the following setup:

Hypothesis 1. We consider a (model, truth, prior) triple $(p(y|x,w), q(y|x), \varphi(w))$ associated to the class of feedforward ReLU neural networks $f : \mathbb{R}^N \times W \to \mathbb{R}^M$. We assume that:

- the true conditional distribution $q(y|x)$ is unknown and to be modelled;
- the prior on inputs, $q(x)$, is a known (i.e. not modelled or estimated) distribution;
- the prior on weights, $\varphi(w)$, is a known distribution on a compact space $W \subseteq \mathbb{R}^d$;
- the model is a standard regression model on f; that is, p is multivariate normally distributed of dimension M with mean $f(x,w)$ and identity covariance matrix, so $p(y|x,w) \sim \mathcal{N}(f(x,w), \mathbb{1}_M)$ with model density given by

$$p(y|x,w) = \frac{1}{(2\pi)^{\frac{M}{2}}} \exp\left(-\frac{1}{2}\|y - f(x,w)\|^2\right),$$

  where $\|.\|^2$ is the standard Euclidean norm on the output space $\mathbb{R}^M$.

Thus we can express the joint densities in terms of conditional densities,

$$q(y,x) = q(y|x)q(x), \quad \text{and} \quad p(y,x|w) = p(y|x,w)q(x).$$

Remark 8. The case in which $q(x)$ is to be modelled is of great interest in many real world settings such as natural language processing or image generation. Indeed, *generative models* are the operative phrase here, where the objective is to train a network to generate data similar to its inputs (see (*Generative Models* 2016, June 16) for more examples and explanation). Our hypothesis on $q(x)$ is valid for the purposes of this thesis due to the nature of our experiments in Chapter 4.

2.2.2 The Kullback-Leibler divergence $K(w)$

In order to train a model, we may aim to minimise a loss function that measures the "distance" between the true distribution and the model for a given w. Note that this is *not* necessarily always the statistical objective we are interested in, for example we may wish to minimise other measures such as the expected generalisation error, but it is the one we shall begin with.

Definition 4. The *entropy* S of the true conditional distribution $q(y|x)$ is

$$S = \mathbb{E}_{D_n}[-\log q(y|x)] = -\iint_{\mathbb{R}^{N+M}} q(y,x)\log q(y|x)dxdy.$$

The *negative log loss* (or *negative log likelihood*) $L(w)$ of a model for a given $w \in W$ is

$$L(w) = \mathbb{E}_{D_n}[-\log p(y|x,w)] = -\iint_{\mathbb{R}^{N+M}} q(y,x)\log p(y|x,w)dxdy.$$

Remark 9. We can also define the joint entropy $S_J = -\iint q(y,x)\log q(y,x)dydx$ and the input entropy $S_x = -\iint q(y,x)\log q(x)dxdy$, thus $S_J = S + S_x$, but since $q(x)$ is assumed known and does not depend on w we are really only interested in the quantity S.

Given arbitrary probability distributions $p(z)$ and $q(z)$, one typically defines the Kullback-Leibler divergence or the *relative entropy* between the two models to be

$$K(q||p) = \int q(z)\log\frac{p(z)}{q(z)}dz.$$

Since our $q(y,x)$ and $p(y,x|w)$ both have the known $q(x)$ as factors, we may refine this definition for our purposes as follows:

Definition 5. The Kullback-Leibler (KL) divergence between the true distribution $q(y,x)$ and the model $p(y,x|w)$ is a function $K : W \to \mathbb{R}$ defined by $K(w) = \mathbb{E}_{D_n}\left[\log \frac{q(y|x)}{p(y|x,w)}\right]$, that is,

$$K(w) := \iint_{\mathbb{R}^{N+M}} q(y|x)q(x) \log \frac{q(y|x)}{p(y|x,w)} dxdy\,. \tag{2.4}$$

It clearly satisfies $K(w) = L(w) - S$.

Though it is often thought of as being a distance, it is not a true metric as it is not symmetric in p and q, nor does it satisfy the triangle inequality **Ref**. It is, however, a loss function, as the next lemma shows.

Lemma 1. *Let $q(y,x)$ and $p(y,x|w) > 0$ be continuous probability density functions. Then $K(w) \geq 0$ for all $w \in W$, and $K(w) = 0$ if and only if $p(y|x,w) = q(y|x)$ for all $x \in \mathbb{R}^N$, $y \in \mathbb{R}^M$.*

Proof. First note that if $q(y,x) = 0$ on some open set $A \subseteq \mathbb{R}^{N+M}$, since $\lim_{x\to 0} x \log x = 0$ we may define in good conscience

$$q(y,x)\log q(y,x) - q(y,x)\log p(y,x|w) := 0.$$

Thus there will be no contribution to $K(w)$ from the region A, so we can assume without loss of generality that $q(y,x) > 0$ on the region of integration.

Consider the real-valued function $S(t) = -\log t + t - 1$ for $t \in (0,\infty)$ which is well defined, continuous and differentiable everywhere on this domain. Then clearly $S(1) = 0$, and indeed we can show that $t = 1$ is the only root. Since $S'(t) = -\frac{1}{t} + 1$, $S(t)$ has a stationary point at $t = 1$, is strictly decreasing on $(0,1)$ and strictly increasing on $(1,\infty)$, thus by continuity we see that $t = 1$ is the only root. Then since $S''(t) = \frac{1}{t^2}$, so $S''(1) = 1 > 0$, we see that S is concave up at $t = 1$, thus showing $S(t) \geq 0$ for all $t \in (0,\infty)$ and $S(t) = 0$ if and only if $t = 1$.

But then since p and q are probability distributions, hence $\iint_{\mathbb{R}^{N+M}} p(y,x|w)dxdy = 1$ and $\iint_{\mathbb{R}^{N+M}} q(y,x)dxdy = 1$, we have

$$\begin{aligned}\iint_{\mathbb{R}^{N+M}} q(y,x)S\left(\frac{p(y,x|w)}{q(y|x)}\right)dxdy &= \iint_{\mathbb{R}^{N+M}} q(y,x)\log\left(\frac{q(y,x)}{p(y,x|w)}\right)dxdy \\ &\quad + \iint_{\mathbb{R}^{N+M}} q(y,x)\frac{p(y,x|w)}{q(y|x)}dxdy - \iint_{\mathbb{R}^{N+M}} q(y,x)dxdy \\ &= K(w)\,.\end{aligned}$$

Since $q(y,x)$, $p(y,x|w) > 0$ we have $0 < \frac{p(y,x|w)}{q(y,x)} < \infty$, hence the integrand in the first integral is non-negative, thus the integral itself is non-negative, so $K(w) \geq 0$.

We have shown that if $p(y,x|w) = q(y,x)$ then $K(w) = 0$, so suppose $K(w) = 0$. Since $S(t) \geq 0$ and $q(y,x) > 0$ are continuous and non-negative on $\mathbb{R}^{N+M}$, by standard real analysis results we must have $S\left(\frac{p(y,x|w)}{q(y,x)}\right) = 0$ for all $(x,y) \in \mathbb{R}^{N+M}$, hence $\frac{p(y,x|w)}{q(y,x)} = 1$ as stated. □

Our statistical learning objective to minimise $K(w)$ thus becomes finding the zero-sets:

Definition 6. The set of *true parameters* is defined as

$$W_0 := \{w \in W \mid K(w) = 0\} = \{w \in W \mid p(y|x,w) = q(y|x)\}\,, \tag{2.5}$$

where the second equality follows from Lemma 1.

We say that the true distribution $q(y|x)$ is *realisable* by the model $p(y|x,w)$ if W_0 is non-empty. That is, there exists a $w \in W$ such that $q(y|x) = p(y|x,w)$.

Remark 10. The definition of realisability should be interpreted as saying that the chosen model is sufficiently expressive to perfectly capture the true distribution in question. This is, of course, unlikely to occur in real world distributions (especially at the scale of datasets at which most deep learning occurs), but the non-realisable case is significantly more technically challenging to deal with, and many of the key results of singular learning theory do not hold under this hypothesis (though they may be generalised with sufficient technicality).

Under Hypothesis 1 we see that when $q(y|x)$ is realisable $K(w)$ is just the mean squared error weighted by the prior on inputs $q(x)$:

Lemma 2. *Let $q(y|x) = p(y|x,w_0)$ be realisable, defined by a true parameter $w_0 \in W$, and define $K(w,w_0)$ to be $K(w)$ with this definition of $q(y|x)$. Then*

$$K(w,w_0) = \frac{1}{2}\int_{\mathbb{R}^N} q(x)\|f(x,w_0) - f(x,w)\|^2\,dx\,. \tag{2.6}$$

Proof. We calculate $K(w,w_0)$ to be

$$\iint_{\mathbb{R}^{N+M}} \frac{q(x)}{(2\pi)^{\frac{M}{2}}} \exp\left(-\frac{1}{2}\|y - f(x,w_0)\|^2\right) \log\left(\frac{\frac{1}{(2\pi)^{\frac{M}{2}}}\exp\left(-\frac{1}{2}\|y-f(x,w_0)\|^2\right)}{\frac{1}{(2\pi)^{\frac{M}{2}}}\exp\left(-\frac{1}{2}\|y-f(x,w)\|^2\right)}\right) dxdy\,.$$

$$= \frac{1}{2(2\pi)^{\frac{M}{2}}}\iint_{\mathbb{R}^{N+M}} q(x)\exp\left(-\frac{1}{2}\|y-f(x,w_0)\|^2\right)\left(\|y-f(x,w)\|^2 - \|y-f(x,w_0)\|^2\right)dxdy\,.$$

Let $u = y - f(x,w_0)$, so $du = dy$, and let $a = f(x,w) - f(x,w_0) \in \mathbb{R}^M$ which is fixed, then $y - f(x,w) = u - a$ and so

$$K(w,w_0) = \frac{1}{2(2\pi)^{\frac{M}{2}}}\int_{\mathbb{R}^N} q(x)K(w,w_0,x)dx\,, \tag{2.7}$$

where for a fixed $x \in \mathbb{R}^N$ we define

$$K(w,w_0,x) = \int_{\mathbb{R}^M} e^{-\frac{1}{2}\|u\|^2}\left(\|u-a\|^2 - \|u\|^2\right)du = \int_{\mathbb{R}^M} e^{-\frac{1}{2}\|u\|^2}\left(-2a\cdot u + \|a\|^2\right)du\,. \tag{2.8}$$

Recall the standard identity $\int_{\mathbb{R}^M} e^{-\frac{1}{2}\|x\|^2}dx = (2\pi)^{\frac{M}{2}}$. For the dot product term we can show that this contribution is zero by induction on the dimension M. The base case for $M = 1$ is simply $\int_{-\infty}^{\infty} a_1 u_1 e^{-\frac{1}{2}u_1^2}du = 0$ since it is an odd integrand over a symmetric domain. For the inductive step, denote $a = (a_1,\ldots,a_M)$ and $u = (u_1,\ldots,u_M)$ and

suppose $\int_{\mathbb{R}^M}(a\cdot u)e^{-\frac{1}{2}\|u\|^2}du = 0$. Then

$$\begin{aligned}
&\int_{\mathbb{R}^M}\int_{-\infty}^{\infty}(a\cdot u + a_{M+1}u_{M+1})e^{-\frac{1}{2}(\|u\|^2+u_{M+1}^2)}du\,du_{M+1}\\
&=\int_{-\infty}^{\infty}e^{-\frac{1}{2}u_{M+1}^2}du_{M+1}\int_{\mathbb{R}^M}(a\cdot u)e^{-\frac{1}{2}\|u\|^2}du\\
&\quad+\int_{-\infty}^{\infty}a_{M+1}u_{M+1}e^{-\frac{1}{2}u_{M+1}^2}du_{M+1}\int_{\mathbb{R}^M}e^{-\frac{1}{2}\|u\|^2}du\,,\\
&=0\,,
\end{aligned}$$

where the first integral vanishes by the inductive hypothesis and the second due to the odd integral over a symmetric domain. Substituting this into (2.8) gives

$$K(w,w_0,x) = \|a\|^2\int_{\mathbb{R}^M}e^{-\frac{1}{2}\|u\|^2}du = (2\pi)^{\frac{M}{2}}\|a\|^2\,,$$

and so recalling the definition of a and substituting into (2.7) yields the result. □

2.2.3 Empirical estimators

In practice, we may only interact with the true distribution $q(y|x)$ by drawing a set of samples $D_n = \{(X_i,Y_i)\}_{i=1}^n$ and calculating an estimator of $K(w)$ based on the observed samples. For notational aesthetics, we let (x_i,y_i) denote the random variables (X_i,Y_i) drawn from $q(y|x)$.

Definition 7. Let $D_n = \{(x_i,y_i)\}_{i=1}^n$ be a dataset of inputs and outputs drawn from the true distribution $q(y|x)$ with associated model $p(y|x,w)$. We define the *empirical entropy* S_n of the true distribution to be

$$S_n := -\frac{1}{n}\sum_{i=1}^n\log q(y_i|x_i)\,, \tag{2.9}$$

the *empirical negative log likelihood* $L_n(w)$ (or *empirical negative log loss*) to be

$$L_n(w) := -\frac{1}{n}\sum_{i=1}^n\log p(y_i|x_i,w)\,, \tag{2.10}$$

and the *empirical Kullback-Leibler divergence* to be

$$K_n(w) := \frac{1}{n}\sum_{i=1}^n\log\frac{q(y_i|x_i)}{p(y_i|x_i,w)} = L_n(w) - S_n\,. \tag{2.11}$$

Remark 11. The negative log likelihood is so-called due to its relation to the likelihood function since

$$e^{-nL_n(w)} = \prod_{i=1}^n p(y_i|x_i,w) = \frac{l(w|x,y)}{\prod_{i=1}^n q(x_i)}\,. \tag{2.12}$$

Lemma 3. *Under Hypothesis 1 $L_n(w)$ has the form*

$$L_n(w) = \frac{M}{2}\log 2\pi + \frac{1}{n}\sum_{i=1}^n\frac{1}{2}\|y_i - f(x_i,w)\|^2\,.$$

Proof. A trivial calculation recalling $p(y_i|x_i,w) = \frac{1}{(2\pi)^{\frac{M}{2}}} \exp\left(-\frac{1}{2}\|y_i - f(x_i,w)\|^2\right)$. □

Lemma 3 means that under Hypothesis 1 we can interpret the negative log likelihood as simply being the *mean-squared error* plus a constant that only depends on M.

These empirical quantities are indeed estimators of their respective non-empirical quantities:

Lemma 4. *The empirical estimators satisfy*

$$\mathbb{E}_{D_n}[K_n(w)] = K(w)\,, \quad \textit{and} \quad \mathbb{E}_{D_n}[S_n] = S\,, \quad \textit{and} \quad \mathbb{E}_{D_n}[L_n(w)] = L(w)\,.$$

If $K(w), S, L(w) < \infty$ then as $n \to \infty$ we have almost sure convergence

$$K_n(w) \xrightarrow{a.s.} K(w)\,, \quad \textit{and} \quad S_n \xrightarrow{a.s.} S\,, \quad \textit{and} \quad L_n(w) \xrightarrow{a.s.} L(w)\,.$$

Proof. We will only calculate $K_n(w)$ as the others are identical. Let $w \in W$ be fixed, then

$$\begin{aligned}\mathbb{E}_{D_n}[K_n(w)] &= \mathbb{E}_{D_n}\left[\frac{1}{n}\sum_{i=1}^n \log\frac{q(y_i|x_i)}{p(y_i|x_i,w)}\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{D_n}\left[\log\frac{q(y_i|x_i)}{p(y_i|x_i,w)}\right] \\ &= \frac{1}{n}\sum_{i=1}^n \int_{\mathbb{R}^{N+M}} q(x,y)\log\frac{q(y|x)}{p(y|x,w)}dxdy = \frac{1}{n}\sum_{i=1}^n K(w) = K(w)\,.\end{aligned}$$

The second statement is a simple corollary of the above calculation by Kolmogorov's law of large numbers (Resnick 1999). □

Remark 12. We can reformulate our definition of the posterior in Definition 2 to be in terms of our empirical estimates, which will later make the analogy to statistical physics clearer. Combining (2.3) and (2.11) gives

$$p(w|D_n) = \frac{1}{Z_n^0}\varphi(w)e^{-nK_n(w)}\,, \quad \text{where} \quad Z_n^0 = \int_W \varphi(w)e^{-nK_n(w)}dw\,.$$

We call Z_n^0 the normalised evidence. Though $K_n(w)$ is the quantity we are most interested in, since $K_n(w) = L_n(w) - S_n$ we may reformulate these definitions to be in terms of $L_n(w)$ since S_n does not depend on w and thus is a factor of $\phi(w)e^{-nL_n(w)}$ and Z_n^0. Morally speaking from the point of view of estimating w, S_n is irrelevant as it is a constant in the limit $n \to \infty$. Thus we may instead write the posterior as

$$p(w|D_n) = \frac{1}{Z_n}\varphi(w)e^{-nL_n(w)}\,, \quad \text{where} \quad Z_n = \int_W \varphi(w)e^{-nL_n(w)}dw\,. \tag{2.13}$$

Then we see that $Z_n^0 = e^{S_n}Z_n$.

As discussed previously, there are other forms of generalisation that we may be interested in, specifically the training and generalisation errors associated to different forms of estimation of the true distribution. In Bayes estimation the true distribution is estimated via the Bayes predictive distribution, $\mathbb{E}_w[p(y|x,w)]$. In Gibbs estimation we draw a single sample $w \sim p(w|D_n)$, hence the associated error will be the average divergence across such samples.

Definition 8. Let $D_n = \{(x_i,y_i)\}_{i=1}^n$ be a dataset of inputs and outputs drawn from the true distribution $q(y|x)$ with associated model $p(y|x,w)$.

The *Bayes training error* is given by

$$B_t = \frac{1}{n}\sum_{i=1}^{n} \log \frac{q(y_i|x_i)}{\mathbb{E}_w[p(y_i|x_i,w)]}\,, \tag{2.14}$$

which is the empirical KL divergence from $q(y,x)$ to the predictive distribution.

The *Bayes generalisation error* is

$$B_g = \mathbb{E}_{D_n}\left[\log \frac{q(y_i|x_i)}{\mathbb{E}_w[p(y|x,w)]}\right]\,, \tag{2.15}$$

the KL divergence from $q(y|x)$ to the predictive distribution.

The *Gibbs training error* is given by

$$G_t = \mathbb{E}_w[K_n(w)]\,, \tag{2.16}$$

which is the mean empirical KL divergence from $q(y|x)$ to $p(y|x,w)$.

The *Gibbs generalisation error* is

$$G_g = \mathbb{E}_w[K(w)]\,, \tag{2.17}$$

the mean KL divergence from $q(y|x)$ to $p(y|x,w)$.

Remark 13. One can use the following identity to rewrite the Bayes errors:

$$\log \frac{q(y|x)}{\mathbb{E}_w[p(y|x,w)]} = -\log \mathbb{E}_w\left[\frac{p(y|x,w)}{q(y|x)}\right]\,.$$

The same proof as in Lemma 4 shows that $\mathbb{E}_{D_n}[B_t] = B_g$ and $\mathbb{E}_{D_n}[G_t] = G_g$.

2.2.4 Tempered posterior

For reasons that will become clear later in drawing our analogy to statistical physics, we introduce a generalised version of the Bayesian posterior:

Definition 9. The *tempered posterior* is defined as

$$p^\beta(w|D_n) := \frac{1}{Z_n^\beta}\varphi(w)e^{-n\beta L_n(w)}\,, \quad \text{where} \quad Z_n^\beta = \int_W \varphi(w)e^{-n\beta L_n(w)}dw\,.$$

We call β the *inverse temperature*. We denote the expectation of some function $f(w)$ with respect to the tempered posterior by

$$\mathbb{E}_w^\beta[f(w)] := \frac{1}{Z_n^\beta}\int_W f(w)\varphi(w)e^{-n\beta L_n(w)}dw\,.$$

Remark 14. Clearly when $\beta = 1$ we have $p^\beta(w|D_n) = p(w|D_n)$, the standard posterior. As such, we will mainly refer to the tempered posterior for the remainder of the thesis, where $\beta = 1$ can be viewed as a special case. Note that $\mathbb{E}_w = \mathbb{E}_w^1$, and other variables with a superscript β shall refer to the tempered posterior, e.g., $G_t^\beta = \mathbb{E}_w^\beta[K_n(w)]$.

The tempered posterior is well motivated in purely mathematical terms too. In computational Bayesian statistics, the presence of β can be used as a tunable hyperparameter. Furthermore, it arises naturally as the object which minimises information complexity under suitable constraints (Zhang 2006), as the next lemma shows.

Lemma 5. *The tempered posterior $p^\beta(w|D_n)$ is the unique maximiser of the relative entropy with respect to the prior $\varphi(w)$, subject to the constraint*

$$\mathbb{E}_{w\sim P}\left[nL_n(w)\right] = \mu_\beta\,,$$

for some $\mu_\beta \in \mathbb{R}$. ***Acknowledge Matt***

Proof. Given the relative entropy functional

$$K(P||\varphi) = -\int_W P(w)\log\frac{P(w)}{\varphi(w)}dw\,,$$

we want to find a probability distribution $P(w)$ that maximises $K(P||\varphi)$ subject to the following constraints:

$$-\sum_{i=1}^n \int_W P(w)\log p(y_i|x_i,w)dw = \mu_\beta\,,\ \text{ and }\ \int_W P(w)dw = 1\,,\ \text{ and }\ \int_W \varphi(w)dw = 1\,,$$

where $\mu_\beta \in \mathbb{R}$ is assumed fixed and given. Let $k(w,P)$ denote the integrand in $K(P||\varphi)$ and let $g_1(w,P), g_2(w,P)$ and $g_3(w,P)$ respectively denote the integrands in the constraints above and let λ_1, λ_2 and λ_3 denote respective Lagrange multipliers. Then we wish to freely optimise the functional

$$\mathcal{F}[\{\lambda_i\}, P(w)] = \int_W \left[k(w,P) - \sum_{j=1}^3 \lambda_j g_j(w,P)\right] dw + \lambda_1\mu_\beta + \lambda_2 + \lambda_3\,. \tag{2.18}$$

We can thus appeal to the Euler-Lagrange equation which states that $\mathcal{F}$ is extremised at the function P such that

$$\frac{d}{dw}\left(\frac{\partial k}{\partial P'}\right) - \frac{\partial k}{\partial P} - \sum_{j=1}^3 \lambda_j\left[\frac{d}{dw}\left(\frac{\partial g_i}{\partial P'}\right) - \frac{\partial g_i}{\partial P}\right] = 0$$

subject to the same constraints as above. This then evaluates to

$$\log\frac{P(w)}{\varphi(w)} + 1 - \lambda_1\sum_{i=1}^n \log p(y_i|x_i,w) + \lambda_2 = 0\,,$$
$$\text{so}\quad P(w) = e^{-(1+\lambda_2)}\varphi(w)\prod_{i=1}^n p^{\lambda_1}(y_i|x_i,w)\,.$$

We can then recognise $e^{-(1+\lambda_2)}$ as the normalising constant with λ_2 solved appropriately, and λ_1 as the inverse temperature β solved according to the other two constraints **doesn't this mean it is not a free parameter?**, thus showing that $P(w) = p^\beta(w|D_n)$. □

Remark 15. When $\beta \to \infty$, the posterior is infinitely concentrated at the maximum likelihood estimators $\hat{w}$. That is, $\lim_{\beta\to\infty} p(w|D_n) = \delta(w - \hat{w})$, where δ is the Dirac delta function. **This intuitively made sense for something like a sharply peaked Gaussian, but for a really flat (i.e. all $y < 1$) Gaussian it's less clear to me. Can I prove this?**

2.2.5 Free energy

The free energy is a fundamental object of study. Primarily, one can think of the free energy as being a measure of inverse posterior density associated to a particular region of W, or in physical terms, a macroscopic state, which we discuss further in Section 2.3. Such a view implies that minimising the free energy is perhaps the *fundamental objective of statistical learning*, and indeed many of the results of Watanabe suggest this is the correct approach to understanding how neural networks generalise so effectively, as well as being the central object describing how neural networks undergo phase transitions. Furthermore, the free energy encodes information about a statistical ensemble in the sense that expectations and variances of quantities of interest, such as $L_n(w)$, arise naturally as derivatives of the free energy.

Definition 10. Given a dataset D_n, we define the *total empirical free energy* $F_n^\beta \in \mathbb{R}$ as

$$F_n^\beta = -\log Z_n^\beta = -\log\left(\int_W \varphi(w) e^{-n\beta L_n(w)} dw\right).$$

Let us inspect this definition a bit closer. F_n^β depends on the choice of model p and prior φ, but more importantly it is inherently a random variable that depends on the random dataset D_n. To investigate the posterior landscape of a given true network in the search for phase transitions we will want to make statements independent of D_n, hence we may instead define the *total free energy* as a function of $\mathbb{E}_{D_n}[nL_n(w)] = nL(w)$,

$$\overline{F}_n^\beta = -\log\left(\int_W \varphi(w) e^{-n\beta L(w)} dw\right).$$

Note that $\overline{F}_n^\beta$ still depends on n even though the randomness in D_n has been marginalised out. Indeed, it is stated (without proof) in Watanabe 2018 that F_n^β and $\overline{F}_n^\beta$ are asymptotically equivalent up to constant order, meaning we may interchange statements about either. **Clearly the lack of proof in Watanabe bugs me. I'm sure I probably can't prove this, but how can I make this claim stronger?**

As explained in the opening, our main interest in the free energy is as a measure of posterior density in regions of W.

F_n^β depends on the choice of model p and prior φ, but is also a function of n, β and the dataset D_n taking values in $\mathbb{R}^{N+M}$. In physics literature it is typically thought of as depending on macroscopic observables, such as the volume V or temperature T, where phase transitions **(section 4)** occur at critical values of these macroscopic observables. Motivated by this perspective, we are lead to define a projected version of the free energy in which the integration takes place over regions of W that correspond to fixed values of a given macroscopic observable.

Definition 11. Let $V : W \to \mathbb{R}$ be an analytic function. The *V-projected free energy* is defined as

$$F : \mathbb{N} \times \mathbb{R}_{\geq 0} \times \mathbb{R} \longrightarrow \mathbb{R}$$
$$F(n, \beta, v) = \mathbb{E}_{D_n}\left[-\log\left(\int_{\{w|V(w)=v\}} \varphi(w) e^{-n\beta L_n(w)} dw\right)\right].$$

If we are averaging over D_n then I don't believe F is a function of n - surely this is, implicitly, the random variable being marginalised out? That said, if I put the expectation in the exponent then I would have $\mathbb{E}_{D_n}[nL_n(w)] = nL(w)$ so I guess the n is still there...

Remark 16. **Make the remark about taking expectation inside the integral.**

In order to demonstrate phase transitions in section blah, we require an estimation method for $F(n, \beta, v)$, which we now develop.

Lemma 6. *The free energy satisfies*

$$\frac{\partial F_n^\beta}{\partial \beta} = \mathbb{E}_w^\beta[nL_n(w)] = nG_t^\beta + \mathbb{E}_w^\beta[S_n]\,,$$
$$\text{and} \quad \frac{\partial^2 F_n^\beta}{\partial \beta^2} = -\mathbb{E}_w^\beta[(nL_n(w))^2] + \mathbb{E}_w^\beta[nL_n(w)]^2 = -\mathbb{V}[nL_n(w)]\,,$$

where $\mathbb{V}$ *denotes the variance.*

Proof. For the first result, by straight calculation we have

$$\frac{\partial F_n^\beta}{\partial \beta} = -\frac{1}{Z_n^\beta}\frac{\partial Z_n^\beta}{\partial \beta} = -\frac{1}{Z_n^\beta}\frac{\partial}{\partial \beta}\left(\int_W \varphi(w)e^{-n\beta L_n(w)}dw\right) = -\frac{1}{Z_n^\beta}\int_W \varphi(w)\frac{\partial}{\partial \beta}e^{-n\beta L_n(w)}dw$$
$$= \frac{1}{Z_n^\beta}\int_W nL_n(w)\varphi(w)e^{-n\beta L_n(w)}dw = \mathbb{E}_w^\beta[nL_n(w)]\,.$$

The second equality here follows from the definition G_t^β and $L_n(w) = K_n(w) + S_n$. For the second derivative we have

$$\frac{\partial^2 F_n^\beta}{\partial \beta^2} = -\frac{\partial}{\partial \beta}\left(\frac{1}{Z_n^\beta}\frac{\partial Z_n^\beta}{\partial \beta}\right) = -\left(\frac{\partial}{\partial \beta}\frac{1}{Z_n^\beta}\right)\frac{\partial Z_n^\beta}{\partial \beta} - \frac{1}{Z_n^\beta}\frac{\partial^2 Z_n^\beta}{\partial \beta^2}$$
$$= \left(\frac{1}{Z_n^\beta}\frac{\partial Z_n^\beta}{\partial \beta}\right)^2 - \frac{1}{Z_n^\beta}\int_W (nL_n(w))^2\varphi(w)e^{-n\beta L_n(w)}dw = \mathbb{E}_w^\beta[nL_n(w)]^2 - \mathbb{E}_w^\beta[(nL_n(w))^2]\,.$$

□

Lemma 7. *Assume that* $L_n(w)$ *is not a constant in* w*. Denote* $F_n^\beta = F_n(\beta)$*. Then*

1. $\mathbb{E}_w^\beta[nL_n(w)]$ *is a decreasing function of* β*.*
2. *There exists a unique* $\beta^* \in (0,1)$ *satisfying*

$$F_n(1) = \mathbb{E}_w^{\beta^*}[nL_n(w)]\,. \tag{2.19}$$

Proof. From Lemma 6, since $\frac{\partial}{\partial \beta}\mathbb{E}_w^\beta[nL_n(w)] = -\mathbb{V}[nL_n(w)]$, and the variance is always positive by the Cauchy-Schwarz inequality, we see that $\frac{\partial}{\partial \beta}\mathbb{E}_w^\beta[nL_n(w)] < 0$ showing the first claim.

For the second claim, first note that by definition $F_n(0) = 0$, hence

$$F_n(1) = \int_0^1 F_n'(\beta)d\beta\,. \tag{2.20}$$

Further, $F_n(\beta)$ is a continuous function in β since the integral is independent of β and the integrand is a continuous function of β for any sufficiently well behaved $L_n(w)$, of which ours are by hypothesis. Thus, by the mean value theorem there exists a unique $\beta^* \in (0,1)$ such that

$$F_n'(\beta^*) = \mathbb{E}_w^{\beta^*}[nL_n(w)] = \frac{F_n(1) - F_n(0)}{1-0} = F_n(1)\,. \tag{2.21}$$

□

Would like to show that FE is only minimised at points on W_0. How do I prove this?

2.3 Deep learning as a Gibbs ensemble

Based on our discussions up until this point which have included phrases such as entropy, free energy, partition function, and more, it should not be surprising to learn that we can formulate all of the above concepts into a statistical physics context. Indeed, the goal of this section is to explain an interpretation of our statistical learning setup as a physical canonical ensemble. This analogy is by no means complete, but we hope it may serve as a useful conceptual framework for physical intuition, and also as a direct mathematical analogy to the widely developed literature of statistical physics.

Thermodynamics is the study of macroscopic observables, such as energy, volume, pressure and mole numbers, associated to equilibrium states. Its central problem is to predict the equilibrium state which eventually results after a change to the total system, for example, after a constraint is removed from a system. We refer the reader to Callen for an extensive overview of this field.

We formulate the learning machine as a *Gibbs ensemble*, in which we imagine the learning process as a physical system in contact with a thermal reservoir that allows an exchange in energy, which we will see corresponds to the ability to cause statistical fluctuations of the system. In physics literature, the Gibbs ensemble, also known as the *canonical ensemble*, is defined as follows

Definition 12. Consider a system with d particles, in a box of volume V, weakly coupled to and in thermal equilibrium with an infinitely large heat reservoir at absolute temperature T. The number of particles in the system is fixed but heat is exchanged with the environment to maintain a temperature T. Let $\Gamma \subseteq \mathbb{R}^d$ denote the configuration space (or phase space), where $\sigma \in \Gamma$ is a microscopic state (i.e. configuration) of the system of particles. To each microscopic state is associated an energy given by the *Hamiltonian*, denoted $H(\sigma)$. The fundamental postulate of the ensemble is that the probability density of points in phase space $\rho : \Gamma \to \mathbb{R}$ is given by the *Gibbs* (or *Boltzmann*) distribution,

$$\rho(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z_d}\,, \quad \text{where} \quad Z_d = \int_\Gamma e^{-\beta H(\sigma)} d\Gamma\,. \tag{2.22}$$

We can immediately make the identification between the phase space Γ of microscopic states and our weight space W, whereby a microstate configuration $\sigma \in \Gamma$ is identified with a weight $w \in W$ (and we assume that the measure on Γ assigns an identical probability to each configuration, thus $d\Gamma = d\sigma = dw$). Comparing the definition of $\rho(\sigma)$ in (2.22) to $p^\beta(w|D_n)$ in Definition 9, we note that to identify the Hamiltonian of the system we should rewrite

$$\begin{aligned} p^\beta(w|D_n) &= \frac{1}{Z_n^{0,\beta}} \exp\Big(-\beta(nK_n(w) - \frac{1}{\beta}\log\varphi(w)\Big) \\ \text{with} \quad Z_n^{0,\beta} &= \int_W \exp\Big(-\beta(nK_n(w) - \frac{1}{\beta}\log\varphi(w)\Big) dw\,, \end{aligned}$$

which thus leads us to define:

Definition 13. Given a dataset D_n, the (random) *Hamiltonian* of our learning machine is

$$H_n(w) := nK_n(w) - \frac{1}{\beta}\log\varphi(w)\,, \tag{2.23}$$

where $\beta = \frac{1}{T}$ is the inverse temperature of the ensemble.

The Hamiltonian thus measures the violation of two different penalty terms and can be thought of as our cost function which we wish to minimise. The primary violation is the loss of the model compared to the true distribution, $nK_n(w)$, whereas the secondary violation is imposed by the prior term $-\frac{1}{\beta}\log\varphi(w)$. Notice that if $\varphi(w)$ vanishes (or is negligibly small) then $H_n(w)$ will be very large, meaning we can interpret $\varphi(w)$ as walls containing a gas. The posterior will thus be concentrated in those regions of W with low Hamiltonian values, which is where $nK_n(w)$ is small and $\varphi(w)$ is large.

The physical analogy also provides insight into the definition of the free energy in Definition 10. The partition function can be viewed as the sum of Boltzmann weights over all possible configurations, meaning one might hope to express an "effective" Hamiltonian F such that

$$e^{-\beta F} \approx \sum_W e^{-\beta H(w)} dw\,.$$

We thus see that the free energy F is defined to be the effective Hamiltonian that makes this work.

Using this setup, we can define macroscopic thermodynamic parameters that are implicit in statistical learning theory. The *average energy* of the system is given by $U = \mathbb{E}_w^\beta[H_n(w)]$, where we have

$$\begin{aligned} U &= \int_W p^\beta(w|D_n) H_n(w) dw = \int_W p^\beta(w|D_n)\left[nK_n(w) - \frac{1}{\beta}\log\varphi(w)\right] \\ &= n\mathbb{E}_w^\beta[K_n(w)] + \frac{1}{\beta}\mathbb{E}_w^\beta[-\log\varphi(w)]\,. \end{aligned}$$

We immediately recognise the first term as being $nG_t(n,\beta)$, the Gibbs training error seen in the previous section **(fix)**. The other term is... **need to think about what this term is. Expectations of arbitrary terms are somehow a bit funny to me.**

Further, we have the entropy of the Boltzmann distribution as $S = \mathbb{E}_w^\beta[-\log(p^\beta(w|D_n)]$, hence

$$S = -\int_W p^\beta(w|D_n)\log(p^\beta(w|D_n) = n\beta\mathbb{E}_w^\beta[K_n(w)] + \mathbb{E}_w^\beta[-\log\varphi(w)] - \mathbb{E}_w^\beta[F_n^{0,\beta}]\,.$$

Need to interpret these terms.

Put in original paragraph about derivatives of free energy in canonical ensemble and how this leads to general susceptibilities.

Given a physical ensemble, the free energy as a function of thermodynamic quantities is of fundamental importance since it contains information about other (adjective - extensive?) quantities in the sense that such quantities arise as derivatives of the free energy. For example, the free energy F in the canonical ensemble with the number of particles N and the temperature T fixed, allows us to derive quantities such as the entropy, $S = -\frac{\partial F}{\partial T}$, the specific heat capacity (at constant volume) $C_V = -T\frac{\partial^2 F}{\partial T^2}$, and the pressure $P = -\frac{\partial F}{\partial V}$.

2.4 Basic results of Singular Learning Theory

Chapter 3

Symmetries of W_0

The aim of this chapter is to fully characterise the symmetries of W_0 for a given realisable model under particular conditions. We begin with the minimal case, whereby there is no possible overparamaterisation of the network, and then extend this to reducible cases thereafter.

3.1 Single output two layer feedforward ReLU networks

We begin by analysing two layer feedforward ReLU neural networks with one put, so $L = 2$ and $M = 1$, and let d denote the width of the hidden layer. Let $f_w : \mathbb{R}^N \to \mathbb{R}$ be such a function. For the sake of readability we let w and b denote the weights and biases associated to the first layer, and q and c denote the weights and biases associated to the second layer, thus we have

$$f_w(x) = c + \sum_{i=1}^{d} q_i \mathrm{ReLU}(\langle w_i, x\rangle + b_i) \tag{3.1}$$
$$\text{where} \quad w = (w_1, \ldots, w_d, b_1, \ldots, b_d, q_1, \ldots, q_d, c) \in W \subseteq \mathbb{R}^{4d+1},$$

where for each $i = 1, \ldots, d$ we have $w_i \in \mathbb{R}^2$, and $b_i, q_i, c \in \mathbb{R}$. Let H_i denote the activation boundaries (where the output of ReLU switches from being zero to non-zero) associated to f_w, where

$$H_i = \{x \in \mathbb{R}^N \mid \langle w_i, x\rangle + b_i = 0\}. \tag{3.2}$$

We say H_i is non-trivial if $w_i \neq 0$, in which case each H_i is a hyperplane in $\mathbb{R}^N$, and when $N = 2$ they are merely lines in the plane. We can make sense of these boundaries in the context of foldsets:

Definition 14. Let $U \subseteq \mathbb{R}^N$ be open and $f : U \to \mathbb{R}$ a continuous piecewise linear function. The *foldset* of f is

$$\mathcal{F}(f) = \left\{ x \in U \;\middle|\; f \text{ is not differentiable at } x \right\}.$$

Lemma 8. *Let $f_w : \mathbb{R}^N \to \mathbb{R}$ be a two layer feedforward ReLU neural network as in (3.1). Then f_w is continuous, and*

$$\mathcal{F}(f_w) = \bigcup_{i=1}^{d} \{H_i \mid w_{\bullet,i} \neq 0\}. \tag{3.3}$$

Proof. Since $\mathrm{ReLU}(x)$ is continuous and f_w is a sum of ReLU's composed with affine functions, continuity is clear. Note that $\mathrm{ReLU}(x)$ is non-differentiable at $x = 0$ since for $x \neq 0$ we have

$$\frac{d}{dx}\mathrm{ReLU}(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

which is clearly discontinuous at $x = 0$. If for any i (possibly many) we have $w_i = 0$ then the contribution of this node to $f_w(x)$ will only be $q_i\mathrm{ReLU}(b_i)$ which is a constant. Since differentiation is linear, f_w is non-differentiable on the boundaries that are inputs to the ReLU functions, the H_i, as claimed. □

ReLU neural nets are piecewise affine functions, thus comprise a finite collection of regions with constant gradient in between the foldsets H_i that define the activation boundaries.

Definition 15. Let $\alpha \in \Lambda$ where Λ is an index set. A *domain* $U^\alpha \subseteq \mathbb{R}^N$ of f_w is a connected open set such that: f is a simple plane with constant gradient when f is restricted to U^α, that is,

$$f_w|_{U_\alpha}(x) = \langle w^\alpha, x\rangle + b^\alpha$$

for some w^α and b^α, which are the sum of the weights and biases that are active in the region U^α, and; U^α is the maximal such set for which the previous property holds. Note that $|\Lambda|$ is finite, and by definition $\bigcup_{\alpha\in\Lambda} U^\alpha = \mathbb{R}^2 \setminus (\mathcal{F}(f_w))$.

Remark 17. Note that w^α and b^α also absorb the gradients q_i and bias c. Further, the precise size of Λ is highly non-trivial in general, though the reader is referred to **Ivanov paper** for an interesting discussion when these are lines in the plane for $N = 2$.

3.2 Classification of W_0 for two layer distinguished minimal networks

Let $w^{(0)} \in W$ be a fixed parameter defining a two layer feedforward ReLU neural network $f_0(x) := f(x, w^{(0)})$ with two inputs and one output. Recall from Definition 6 that we have

$$W_0 = \{w \in W \,|\, p(y|x, w) = q(y|x)\}\,.$$

In classifying W_0 we are obviously only interested in the case that it is non-empty, hence we may assume that $q(y|x) = p(y|x, w^{(0)})$. Since probability distributions are uniquely defined (up to a set of measure zero), this condition implies that $p(y|x, w)$ and $p(y|x, w^{(0)})$, under Hypothesis 1, must have the same mean. Thus

$$W_0 = \{w \in W \,|\, f(x, w) = f(x, w^{(0)})\}\,,$$

and so the task of classifying W_0 becomes classifying functional equivalence of this class of networks.

We begin in the simplest case where f_0 is distinguished and minimal.

3.2.1 Definitions and hypotheses

Definition 16. Let f_w and f'_w be two layer feedforward ReLU neural networks with respective widths d and d' in the hidden layer such that $d' \leq d$ and $f_w = f_{w'}$. We say f_w is *minimal* if necessarily $d' = d$.

We say f_w is *distinguished* if each non-trivial activation boundary H_i in (3.2) is unique, that is, $H_i \neq H_j$ for $i \neq j$.

To see why we require the distinguished condition, observe that if $f(x, w)$ has d unique activation boundaries then it is necessarily minimal, but the converse is not necessarily true:

Example 2. Consider $w = ((1, 1), (-1, -1), 0, 0, 1, 1, 0)$, so

$$f(x, w) = \mathrm{ReLU}(x_1 + x_2) + \mathrm{ReLU}(-x_1 - x_2)\,.$$

Then H_1 is the line $x_1 + x_2 = 0$ and H_2 is the line $-x_1 - x_2 = 0$, and so clearly $H_1 = H_2$ as subsets of $\mathbb{R}^2$, thus there is only one unique foldset. However $f(x, w)$ is minimal: suppose there was a one-node neural network $f(x, r) = c + q\mathrm{ReLU}(\langle x, r\rangle + b)$ which produced the same input output map. Then this network necessarily has a region of inactivation by the definition of ReLU, $\langle x, r\rangle + b < 0$, and $f(x, r)$ has zero gradient in this region. But $f(x, w)$ has non-zero gradient in both regions $\{(x_1, x_2) \,|\, x_2 \geq x_1\}$ and $\{(x_1, x_2) \,|\, x_2 \leq x_1\}$, thus we could not have $f(x, r) = f(x, w)$.

Remark 18. One can show that n non-parallel lines divide the plane into $L_n = \frac{n(n+1)}{2} + 1$ different regions. As such, one can in principle check for the minimality of a given network f by counting the number of different piecewise regions and comparing this to L_n.

Armed with these formulations we are thus ready to begin investigating the symmetries of W_0. The results in this section are formulated based on the following assumptions which we summarise for clarity.

Hypothesis 2. Let $q(y|x)$ be a realisable true distribution defined by a two layer feedforward ReLU neural network $f_0 := f(., w^{(0)}) : \mathbb{R}^2 \to \mathbb{R}^1$ of width m for some fixed parameter $w^{(0)} \in W_0$. Let $p(y|x, w)$ be the model as in Hypothesis 1 defined by a two layer feedforward ReLU neural network $f : \mathbb{R}^2 \times W \to \mathbb{R}^1$ of width d. We further impose:

- $f_0(x)$ is minimal and distinguished.
- The width of the true network and the learner are equal, so $m = d$.
- Without loss of generality, $w^{(0)}$ is defined by

$$w^{(0)} = (w_1^{(0)}, \ldots, w_m^{(0)}, b_1^{(0)}, \ldots, b_m^{(0)}, 1, \ldots, 1, 0)\,,$$

  where each $w_i^{(0)} \in \mathbb{R}^2 \in \{(0, 0)\}$ and $b_i^{(0)} \in \mathbb{R}$ for each $1 \leq i \leq d$.

To characterise W_0, we thus set $f(x, w) = f_0(x)$ and describe the possible values of w.

Remark 19. Clearly $f(x, w)$ must also be minimal and distinguished.

3.2.2 Lemmas

The first form of symmetry is a simple permutation of nodes:

Lemma 9. *Let H_i denote the activation boundaries associated to $f(x, w)$ as in (3.2) and let $H_i^{(0)}$ denote the activation boundaries associated to $f_0(x)$. Then there exists some permutation $\sigma \in S_m$ (the symmetric group of order m) such that $H_i = H_{\sigma(i)}^{(0)}$ for each $1 \leq i \leq m$.*

Proof. **Need to fix this proof - is currently wrong, haven't got around to it yet.**

Since $f(x, w) = f_0(x)$ for all x, they must also be non-differentiable at the same points, so $\mathcal{F}(f) = \mathcal{F}(f_0)$. Since the $\mathcal{A}(f) = \mathcal{F}(f)$ for feedforward ReLU networks, we therefore have

$$\mathcal{A}(f) = \mathcal{F}(f(x, w)) = \bigcup_{i=1}^{d} H_i = \bigcup_{j=1}^{m} H_j^{(0)} = \mathcal{F}(f_0(x)) = \mathcal{A}(f_0)\,.$$

Since each H_i and $H_j^{(0)}$ are distinct by hypothesis, we must therefore have a unique $1 \leq j \leq m$ such that $H_i = H_j^{(0)}$. To see this, first observe that there can only be finitely many intersection points of the lines H_i (and similarly for $H_j^{(0)}$), thus let $x \in \mathcal{A}(f)$ be such that it is uniquely associated to a line H_i for some i. Then since the points of intersection must be same for $H_j^{(0)}$ and $\mathcal{A}(f) = \mathcal{A}(f_0)$, x is also uniquely associated to a line $x \in H_j^{(0)}$ for some j. Since x was arbitrary, the lines H_i and $H_j^{(0)}$ must be equal up to points of intersection, but since they are continuous and differentiable we can simply conclude that $H_i = H_j^{(0)}$ for this said j.

Finally, since $m = d$, our previous statement simply says that there is a unique bijection $\sigma : \{1, \ldots, d\} \to \{1, \ldots, m\}$ associating lines in $\mathcal{A}(f_0)$ to lines in $\mathcal{A}(f)$, thus giving the desired $\sigma \in S_m$. □

Next we have scaling symmetry:

Lemma 10. *Let $w, w' \in \mathbb{R}^2 \setminus \{0\}$ and $b, b' \in \mathbb{R}$ be given and let*

$$H = \{x \in \mathbb{R}^2 \,|\, \langle w, x\rangle + b = 0\}, \quad \text{and} \quad H' = \{x \in \mathbb{R}^2 \,|\, \langle w', x\rangle + b' = 0\}\,.$$

Then $H = H'$ if and only if there exists some scalar $\lambda \in \mathbb{R} \setminus \{0\}$ such that $w = \lambda w'$ and $b = \lambda b'$.

Proof. The first direction is simple: suppose $\lambda \in \mathbb{R} \setminus \{0\}$ is such that $w = \lambda w'$ and $b = \lambda b'$, then if $x \in H'$ we have

$$0 = \langle w', x\rangle + b' = \langle \lambda w, x\rangle + \lambda b = \lambda\left(\langle w, x\rangle + b\right)$$

and so dividing by λ shows that $x \in H$, and by symmetry we clearly have $H' \subseteq H$ too, so $H = H'$.

Now suppose $H = H'$. Let $t \in H$ be a scalar multiple of w, so $t = \mu w$ for some $\mu \in \mathbb{R}$, then

$$0 = \langle w, t\rangle + b = \mu\langle w, w\rangle + b\,, \quad \text{so} \quad \mu = -\frac{b}{\langle w, w\rangle}\,, \tag{3.4}$$

and so t is the unique point such that $b = -\langle w, t\rangle$. Similarly we have a unique $t' = \mu' w'$ for $\mu' = -\frac{b'}{\langle w', w'\rangle}$ and $b' = -\langle w', t'\rangle$. Then saying $x \in H$ is now equivalent to $\langle w, x - t\rangle = 0$,

but since $x \in H'$ as well we also have $\langle w', x - t' \rangle = 0$. Taking $x = t'$ in the first case and $x = t$ in the second case, noting $\langle w, t' - t \rangle = -\langle w, t - t' \rangle$ we have a system

$$A_w(t - t') := \begin{pmatrix} w_1 & w_2 \\ w_1' & w_2' \end{pmatrix} \begin{pmatrix} t_1 - t_1' \\ t_2 - t_2' \end{pmatrix} = 0\,, \tag{3.5}$$

thus either $t = t'$ or $\operatorname{rank}(A_w) = 1$ (w and w' are nonzero by hypothesis, excluding the possibility of $\operatorname{rank}(A_w) = 0$). In the first case we have $t = \mu w = \mu' w' = t'$, thus we can take $\lambda = \frac{\mu}{\mu'}$. In the second case, $\operatorname{rank}(A_w) = 1$ implies w and w' are linearly dependent, thus $w = \lambda w'$ for some $\lambda \in \mathbb{R}$. For such a λ we thus have

$$b = -\langle w, t \rangle = -\langle w, t' \rangle = -\lambda \langle w', t' \rangle = \lambda b'\,, \tag{3.6}$$

where the second equality follows from $\langle w, t - t' \rangle = 0$, thus proving the claim. □

Remark 20. The proof of Lemma 10 is easily generalised to hyperplanes in $\mathbb{R}^n$, which we leave as an exercise for the reader.

Lemma 11. *For $f(x, w) = f_0(x)$, there exists a unique $\sigma \in S_m$, and for each $1 \leq i \leq m$ there exists an $\epsilon_i \in \mathbb{Z}_2$ such that*

$$w_i = (-1)^{\epsilon_i} \frac{q_i^{(0)}}{q_i} w_{\sigma(i)}^{(0)}\,, \quad \textit{and} \quad b_i = (-1)^{\epsilon_i} \frac{q_i^{(0)}}{q_i} b_{\sigma(i)}^{(0)}\,. \tag{3.7}$$

Proof. Lemma 9 gives the permutation $\sigma \in S_m$ relating activation boundaries. For each w_i Lemma 10 gives us a $\mu_i \in \mathbb{R}$, which we can decompose into $\mu = (-1)^{\epsilon_i} \lambda_i$ for some $\epsilon_i \in \mathbb{Z}_2$ and $\lambda_i \in \mathbb{R}_{>0}$, so we can initially write

$$w_i = (-1)^{\epsilon_i} \lambda_i w_{\sigma(i)}^{(0)}\,, \quad \text{and} \quad b_i = (-1)^{\epsilon_i} \lambda_i b_{\sigma(i)}^{(0)}\,. \tag{3.8}$$

Let i be fixed and let $x \in H_i \setminus \left(\bigcup_{j \neq i} H_j \right)$, thus excluding intersection points. Let $U \ni x$ be a sufficiently small open ball around x that doesn't intersect any other H_j for $j \neq i$. Recall that by hypothesis $H_i \neq H_j$ for any other $j \neq i$, thus excluding the possibility of any other node being activated across the boundary H_i. Then we can write $U = U^- \cup U^+$ where $U^- = U \cap \{U^\alpha | \text{node } i \text{ is inactive}\}$ and U^+ similarly but for where node i is active. Then we have

$$f|_{U^+}(x, w) = f|_{U^-}(x, w) + q_i(\langle w_i, x \rangle + b_i)\,. \tag{3.9}$$

Similarly, consider the same set up for the line $H_{\sigma(i)}^{(0)} = H_i$ associated to $f_0(x)$, where U_0 is the sufficiently small neighbourhood and U_0^- and U_0^+ are the regions of inactivation and activation respectively. Then we have

$$f_0|_{U^+}(x) = f_0|_{U^-}(x) + q_{\sigma(i)}^{(0)} \left(\langle w_{\sigma(i)}^{(0)}, x \rangle + b_{\sigma(i)}^{(0)} \right)\,.$$

First suppose $\epsilon_i = 0$, so w_i and $w_{\sigma(i)}^{(0)}$ are oriented in the same direction and so $U^+ = U_0^+$. Since $f(x, w) = f_0(x)$ we thus have

$$\begin{aligned} q_i(\langle w_i, x \rangle + b_i) &= f|_{U^+}(x, w) - f|_{U^-}(x, w) \\ &= f_0|_{U^+}(x) - f_0|_{U^-}(x) = q_{\sigma(i)}^{(0)} \left(\langle w_{\sigma(i)}^{(0)}, x \rangle + b_{\sigma(i)}^{(0)} \right)\,, \end{aligned}$$

and so by comparing coefficients we must have

$$w_i = \frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)}\,, \quad \text{and} \quad b_i = \frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)}\,.$$

If $\epsilon_1 = 1$ then w_i and $w^{(0)}_{\sigma(i)}$ are oriented in different directions, thus $U^+ = U_0^-$ and $U^- = U_0^+$ so we have

$$\begin{aligned} q_i\left(\langle w_i, x\rangle + b_i\right) &= f|_{U^+}(x,w) - f|_{U^-}(x,w) \\ &= f_0|_{U^-}(x) - f_0|_{U^+}(x) = -q^{(0)}_{\sigma(i)}\left(\langle w^{(0)}_{\sigma(i)}, x\rangle + b^{(0)}_{\sigma(i)}\right) \end{aligned}$$

and so again comparing coefficients we have

$$w_i = -\frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)}\,, \quad \text{and} \quad b_i = -\frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)}\,.$$

Combining these we can thus write

$$w_i = (-1)^{\epsilon_i}\frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)}\,, \quad \text{and} \quad b_i = (-1)^{\epsilon_i}\frac{q^{(0)}_{\sigma(i)}}{q_i} w^{(0)}_{\sigma(i)} \tag{3.10}$$

as advertised. Reconciling this with (3.8) we see that $\lambda_i = \frac{q_i^{(0)}}{q_i}$ must be positive. □

Remark 21. Note that since we have assumed $q_i^{(0)} = 1$ we have that $\lambda_i = \frac{1}{q_i}$ and so q_i must be positive.

At first glance, one may assume that all orientations must be preserved, that is, all $\epsilon_i = 0$, in order for things to work out. But as the next example shows, this is not necessarily the case.

Example 3. Consider a simple one dimensional ReLU neural network $f : \mathbb{R} \to \mathbb{R}$ with $d = 2$ hidden nodes, then if we define

$$\begin{aligned} f_0(x) &= \mathrm{ReLU}(x+1) + \mathrm{ReLU}(-x+1)\,, \\ \text{and} \quad f(x,w) &= 2 + \mathrm{ReLU}(-x-1) + \mathrm{ReLU}(x-1)\,, \end{aligned}$$

we see that $f(x,w) = f_0(x)$ and so $w = (-1,1,-1,-1,1,1,2)$ is also a true parameter for $w^{(0)} = (1,-1,1,1,1,1,0)$. Notice that here we have $w_1 + b_1 = -(w_1^{(0)} + b_1^{(0)})$ and similarly $w_2 + b_2 = -(w_2^{(0)} + b_2)$, meaning $\epsilon_1 = \epsilon_2 = 1$. But this works because $f_0(x)$ is constant on the domain $U = (-1,1)$.

We use this example as the inspiration for the orientation reversing symmetry:

Lemma 12. *Let* $E = \{i = 1,\dots,d \,|\, \epsilon_i = 1\}$. *Then* $\sum_{i\in E} w^{(0)}_{\sigma(i)} = 0$.

Proof. Let U_α be one of the domains of f such that $U_\alpha \bigcap \left(\bigcup_i H_i\right) = \varnothing$. For notational convenience, define

$$\delta_i^\alpha := \begin{cases} 1 & \text{if} \quad \langle w_i^{(0)}, x\rangle + b_i^{(0)} \geq 0 \\ 0 & \text{otherwise} \end{cases}\,, \quad \text{and} \quad \overline{\delta_i}^{\alpha} := 1 - \delta_i^\alpha\,, \tag{3.11}$$

thus δ_i^α indicates whether node i of $f_0(x)$ is active in the region U^α, and $\overline{\delta_i^\alpha}$ indicates the converse. Let i be a node of $f(x,w)$ and $\sigma(i)$ the corresponding node of $f_0(x)$ such that $H_i = H^{(0)}_{\sigma(i)}$. Following the result of Lemma 11 we can calculate

$$f(x,w) = c + \sum_{i=1}^{d} q_i \mathrm{ReLU}\left(\langle w_i, x\rangle + b_i\right) = c + \sum_{i=1}^{d} q_i \lambda_i \mathrm{ReLU}\left((-1)^{\epsilon_i}\left(\langle w^{(0)}_{\sigma(i)}, x\rangle + b^{(0)}_{\sigma(i)}\right)\right).$$

In particular, the transformation

$$\mathrm{ReLU}(\langle w_i, x\rangle + b_i) \longmapsto \lambda_i \mathrm{ReLU}\left((-1)^{\epsilon_i}\left(\langle w^{(0)}_{\sigma(i)}, x\rangle + b^{(0)}_{\sigma(i)}\right)\right)$$

shows that if $\sigma(i)$ is active in U^α, then we are in one of two situations: either $\epsilon_i = 0$ and $\delta^\alpha_{\sigma(i)} = 1$, or $\epsilon_i = 1$ and $\overline{\delta}^\alpha_{\sigma(i)} = 1$. Therefore, recalling $w^{(0)}_{\sigma(i)} = (-1)^{\epsilon_i} q_i w_i$ from Lemma 11, we have

$$\sum_i \delta^\alpha_{\sigma(i)} w^{(0)}_{\sigma(i)} = \sum_{i\notin E} \delta^\alpha_{\sigma(i)} q_i w_i + \sum_{i\in E} \overline{\delta}^\alpha_{\sigma(i)} q_i w_i = \sum_{i\notin E} \delta^\alpha_{\sigma(i)} w^{(0)}_{\sigma(i)} + \sum_{i\in E} \overline{\delta}^\alpha_{\sigma(i)} (-w^{(0)}_{\sigma(i)}).$$

But by definition we have

$$\sum_i \delta^\alpha_{\sigma(i)} w^{(0)}_{\sigma(i)} = \sum_{i\notin E} \delta^\alpha_{\sigma(i)} w^{(0)}_{\sigma(i)} + \sum_{i\in E} \delta^\alpha_{\sigma(i)} w^{(0)}_{\sigma(i)},$$

so subtracting one from the other shows that

$$\sum_{i\in E} \left(\delta^\alpha_{\sigma(i)} + \overline{\delta}^\alpha_{\sigma(i)}\right) w^{(0)}_{\sigma(i)} = \sum_{i\in E} w^{(0)}_{\sigma(i)} = 0.$$

□

Then Example 3 shows us that we can expect a similar result for the biases.

Lemma 13. *With E as in Lemma 12, $\sum_{i\in E} b^{(0)}_{\sigma(i)} = c$.*

Proof. The same arguments regarding active nodes in U^α as in Lemma 12 applies, recalling that the bias c is "active" on every domain, so we have

$$\sum_i \delta^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)} = c + \sum_{i\notin E} \delta^\alpha_{\sigma(i)} q_i b_i + \sum_{i\in E} \overline{\delta}^\alpha_{\sigma(i)} q_i b_i = c + \sum_{i\notin E} \delta^\alpha_{\sigma(i)} q_i b_i + \sum_{i\in E} \overline{\delta}^\alpha_{\sigma(i)} (-b^{(0)}_{\sigma(i)}),$$

but again recalling

$$\sum_i \delta^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)} = \sum_{i\notin E} \delta^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)} + \sum_{i\in E} \delta^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)},$$

we thus have

$$\sum_{i\in E} \delta^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)} = c - \sum_{i\in E} \overline{\delta}^\alpha_{\sigma(i)} b^{(0)}_{\sigma(i)},$$

and so $\sum_{i\in E} b^{(0)}_{\sigma(i)} = c$. □

Remark 22. Our convention is to take the empty sum to be zero, so if E is empty then we have $c = 0$.

3.2.3 Main theorem

We have thus arrived at the main theorem of this section having classified all of the symmetries of W_0. We first introduce some notation: we denote

$$X := \left\{(\lambda, q) \in \mathbb{R}_{>0} \,|\, \lambda q = 1\right\}, \quad \text{and} \quad \Upsilon := \left\{ \epsilon : \{1, \dots, d\} \to \mathbb{Z}_2 \;\middle|\; \sum_{\epsilon_i = 1} w_i^{(0)} = 0 \right\},$$

and let $G_\alpha(f(x,w)) \in \mathbb{R}^2$ denote the gradient computed by $f(x,w)$ in the region U^α, and similarly $\mathcal{B}_\alpha(f(x,w)) \in \mathbb{R}$ the bias.

Theorem 14. *Let $f_0 : \mathbb{R}^2 \to \mathbb{R}^1, f_0(x) := f(x, w^{(0)})$ be a distinguished minimal feedforward ReLU neural network with two layers, d hidden nodes and some fixed true parameter $w^{(0)} \in W$. Then there is a bijection*

$$\Psi : X^m \times S_m \times \Upsilon \xrightarrow{\cong} W_0$$

$$\theta = \left((\lambda_i, q_i)_{i=1}^m, \sigma, \epsilon\right) \longmapsto \left(\left((-1)^{\epsilon_i} \lambda_i w_{\sigma(i)}^{(0)}\right)_{i=1}^m, \left((-1)^{\epsilon_i} \lambda_i b_{\sigma(i)}^{(0)}\right)_{i=1}^m, (q_i)_{i=1}^m, \sum_{\epsilon_i = 1} b_i^{(0)} \right).$$

Remark 23. We refer to X as *scaling* symmetry, S_m as *permutation* symmetry and Υ as *orientation reversing* symmetry.

Proof. We first verify that Ψ is well defined, that is, $f(x, \Psi(\theta)) = f_0(x)$ as functions. We compute for a fixed $\theta = ((\lambda_i, q_i)_{i=1}^m, \sigma, \epsilon) \in X^m \times S_m \times \Upsilon$

$$\begin{aligned} f(x, \Psi(\theta)) &= \sum_{\epsilon_i = 1} b_i^{(0)} + \sum_{i=1}^d q_i \mathrm{ReLU}\left(\left\langle (-1)^{\epsilon_i} \lambda_i w_{\sigma(i)}^{(0)}, x \right\rangle + (-1)^{\epsilon_i} \lambda_i b_{\sigma(i)}^{(0)} \right) \\ &= \sum_{\epsilon_i = 1} b_i^{(0)} + \sum_{i=1}^d q_i \lambda_i \mathrm{ReLU}\left((-1)^{\epsilon_i} \left(\left\langle w_{\sigma(i)}^{(0)}, x \right\rangle + b_{\sigma(i)}^{(0)} \right) \right) \\ &= \sum_{\epsilon_i = 1} b_i^{(0)} + \sum_{i=1}^d \mathrm{ReLU}\left((-1)^{\epsilon_i} \left(\left\langle w_{\sigma(i)}^{(0)}, x \right\rangle + b_{\sigma(i)}^{(0)} \right) \right). \end{aligned}$$

Thus we see that $f(x, \Psi(\theta))$ has the same foldsets as $f_0(x)$. It remains to check the gradients and biases in any domain U^α agree. Let $\delta_{\sigma(i)}^\alpha$ be as in Lemma 12 so it refers to active nodes of $f_0(x)$. Then for any region U^α the gradient computed by $f_0(x)$ is

$$\mathcal{G}_\alpha(f_0(x)) = \sum_i \delta_{\sigma(i)}^\alpha w_{\sigma(i)}^{(0)},$$

whereas the gradient computed by $f(x, \Psi(\theta))$ is

$$\begin{aligned} \mathcal{G}_\alpha(f(x, \Psi(\theta))) &= \sum_{i \notin E} \delta_{\sigma(i)}^\alpha w_{\sigma(i)}^{(0)} + \sum_{i \in E} \overline{\delta}_{\sigma(i)}^\alpha (-w_{\sigma(i)}^{(0)}) \\ &= \sum_{i \notin E} \delta_{\sigma(i)}^\alpha w_{\sigma(i)}^{(0)} + \sum_{i \in E} \delta_{\sigma(i)}^\alpha w_{\sigma(i)}^{(0)} = \sum_i \delta_{\sigma(i)}^\alpha w_{\sigma(i)}^{(0)} = \mathcal{G}_\alpha(f_0(x)), \end{aligned}$$

where the second equality followed from our hypothesis on $\epsilon \in \Upsilon$. Similarly, the bias computed by $f_0(x)$ is

$$\mathcal{B}_\alpha(f_0(x)) = \sum_i \delta_{\sigma(i)}^\alpha b_{\sigma(i)}^{(0)}$$